

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: • The ACM Digital Library O The Guide

permission class, declaration

SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: permission class declaration

Found 47,261 of 207,474

Sort results

by

Display results

Ţ relevance expanded form

Save results to a Binder  $\overline{\mathbf{x}}$ 

Try an Advanced Search Try this search in The ACM Guide

Open results in a new window

Search Tips

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 10

Relevance scale

Best 200 shown

Generic ownership for generic Java

Alex Potanin, James Noble, Dave Clarke, Robert Biddle

October 2006 ACM SIGPLAN Notices, Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06, Volume 41 Issue 10

Publisher: ACM Press

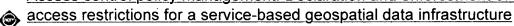
Full text available: pdf(279.65 KB)

Additional Information: full citation, abstract, references, citings, index terms

Ownership types enforce encapsulation in object-oriented programs by ensuring that objects cannot be leaked beyond object(s) that own them. Existing ownership programming languages either do not support parametric polymorphism (type genericity) or attempt to add it on top of ownership restrictions. Generic Ownership provides perobject ownership on top of a sound generic imperative language. The resulting system not only provides ownership quarantees comparable to established systems, but ...

**Keywords**: Java, generics, ownership, type systems

2 Access control policy management: Declaration and enforcement of fine-grained



Andreas Matheus

June 2005 Proceedings of the tenth ACM symposium on Access control models and technologies SACMAT '05

Publisher: ACM Press

Full text available: 📆 pdf(694.91 KB) Additional Information: full citation, abstract, references, index terms

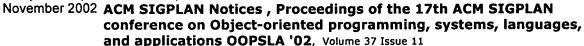
This work describes the declaration and enforcement of geospatial access restrictions for the infrastructure of heterogenous and distributed geospatial information objects, as they are accessible via the service-oriented geospatial data infrastructure (GDI). Assuming a valid XML markup of the objects and their geometry using the Geographic Markup Language (GML), which is an international standard of the Open GIS Consortium, Inc. (OGC), a solution is introduced that allows the declaration and enf ...

Keywords: ACM proceedings, XACML, geospatial, object-based access restriction, spatial access control



Access rights analysis for Java

Larry Koved, Marco Pistoia, Aaron Kershenbaum



Publisher: ACM Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(360.93 KB) terms.

Java 2 has a security architecture that protects systems from unauthorized access by mobile or statically configured code. The problem is in manually determining the set of security access rights required to execute a library or application. The commonly used strategy is to execute the code, note authorization failures, allocate additional access rights, and test again. This process iterates until the code successfully runs for the test cases in hand. Test cases usually do not cover all paths th ...

**Keywords**: Java security, access rights, call graph, data flow analysis, invocation graph, security

Data groups: specifying the modification of extended state



K. Rustan M. Leino

October 1998 ACM SIGPLAN Notices, Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '98, Volume 33 Issue 10

**Publisher: ACM Press** 

Full text available: pdf(1.20 MB)

Additional Information: full citation, abstract, references, citings, index terms

This paper explores the interpretation of specifications in the context of an object-oriented programming language with subclassing and method overrides. In particular, the paper considers annotations for describing what variables a method may change and the interpretation of these annotations. The paper shows that there is a problem to be solved in the specification of methods whose overrides may modify additional state introduced in subclasses. As a solution to this problem, the paper introduc ...

5 Types: Extracting programs from type class proofs



Martin Sulzmann

July 2006 Proceedings of the 8th ACM SIGPLAN symposium on Principles and practice of declarative programming PPDP '06

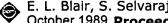
Publisher: ACM Press

Full text available: pdf(250.86 KB) Additional Information: full citation, abstract, references, index terms

Standard presentations of type class translation schemes exhibit some surprising problems when translating Haskell 98 programs. We suggests ways how to fix these problems based on a formal framework for extracting programs from type class proofs. Our description includes type improvement and recursive dictionaries -- something which has not been formally studied before. Thus, we are able to advance the state of art of translating type classes and open up the possibility for new type class applic ...

**Keywords**: overloading, proofs are programs principle, type classes

6 DISC++: A C++ based library for object oriented simulation



E. L. Blair, S. Selvaraj
October 1989 Proceedings of the 21st conference on Winter simulation WSC '89

**Publisher: ACM Press** 



Full text available: pdf(477.43 KB) Additional Information: full citation, abstract, references, citings, index terms

The Object Oriented Programming (OOP) paradigm is generating considerable interest and excitement among systems analysts and programmers concerned with a wide range of applications. This paper presents DISC++ (DIscrete event Simulation in C++), a library of routines written in C and C++ which supports the design and programming of simulation models under both the event scheduling and process interaction world-views. DISC++ allows the simulator to construct simpler mod ...

7 Model driven security: From UML models to access control infrastructures



David Basin, Jürgen Doser, Torsten Lodderstedt

January 2006 ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 15 Issue 1

**Publisher: ACM Press** 

Full text available: pdf(968.83 KB) Additional Information: full citation, abstract, references, index terms

We present a new approach to building secure systems. In our approach, which we call Model Driven Security, designers specify system models along with their security requirements and use tools to automatically generate system architectures from the models, including complete, configured access control infrastructures. Rather than fixing one particular modeling language for this process, we propose a general schema for constructing such languages that combines languages for modeling systems with ...

Keywords: Model Driven Architecture, Object Constraint Language, Role-Based Access Control, Unified Modeling Language, metamodeling, security engineering

8 Software design, languages and systems: Supporting access control policies across





multiple operating systems Lawrence Teo, Gail-Joon Ahn

March 2005 Proceedings of the 43rd annual Southeast regional conference - Volume 2 ACM-SE 43

Publisher: ACM Press

Full text available: pdf(366.71 KB) Additional Information: full citation, abstract, references, index terms

The evaluation of computer systems has been an important issue for many years, as evidenced by the introduction of industry evaluation guides such as the Rainbow Books and the more recent Common Criteria for IT Security Evaluation. As organizations depend on the Internet for their daily operations, the need for evaluation is even more apparent due to new security risks. It is not uncommon for large organizations to evaluate different systems, such as operating systems, to identify which would be ...

**Keywords**: Chameleos, access control, extensibility, flexibility, operating systems, policy specification

Sealed calls in Java packages



Ayal Zaks, Vitaly Feldman, Nava Aizikowitz

October 2000 ACM SIGPLAN Notices, Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '00, Volume 35 Issue 10

Publisher: ACM Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(192.57 KB)

Determining the potential targets of virtual method invocations is essential for interprocedural optimizations of object-oriented programs. It is generally hard to determine such targets accurately. The problem is especially difficult for dynamic languages such as Java, because additional targets of virtual calls may appear at runtime. Current mechanisms that enable inter-procedural optimizations for dynamic languages, repeatedly validate the optimizations at runtime. This paper addresses this ...

Keywords: Java, call devirtualization, call graph, class hierarchy graph, inter-procedural analysis, method inlining, object-oriented programming, sealed package

### 10 Essays in computing science

C. A. R. Hoare January 1989 Book

Publisher: Prentice-Hall, Inc.

Full text available: pdf(20.91 MB) Additional Information: full citation, abstract, references, cited by, review

Charles Antony Richard Hoare is one of the most productive and prolific computer scientists. This volume contains a selection of his published papers. There is a need, as in a Shakespearian Chorus, to offer some apology for what the book manifestly fails to achieve. It is not a complete 'collected works'. Selection between papers of this quality is not easy and, given the book's already considerable size, some difficult decisions as to what to omit have had to be made. Pity the editor weighin ...

### 11 Object-oriented modeling using C++

D. Peter Sanderson, Lawrence L. Rose

January 1988 Proceedings of the 21st annual symposium on Simulation ANSS '88

Publisher: IEEE Computer Society Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(859.30 KB) terms

Object-oriented modeling provides a natural and powerful paradigm for representing the elements of a discrete-state system and their behavior. The concepts of encapsulation and inheritance are central to the realization of the object orientation. The C + + programming language supports encapsulation through the class construct, and inheritance through derived classes. A hierarchy of C + + classes designed to support an event-oriented simulation viewpoint is presented. The use of this packag ...

## 12 <u>Certification of programs for secure information flow</u>

Dorothy E. Denning, Peter J. Denning

July 1977 Communications of the ACM, Volume 20 Issue 7

Publisher: ACM Press

Full text available: pdf(918.82 KB) Additional Information: full citation, abstract, references, citings

ertification mechanism for verifying the secure flow of information through a program. Because it exploits the properties of a lattice structure among security classes, the procedure is sufficiently simple that it can easily be included in the analysis phase of most existing compilers. Appropriate semantics are presented and proved correct. An important application is the confinement problem: The mechanism can prove that a program cannot cause supposedly nonconfidential results to depend on conf ...

**Keywords**: confinement, information flow, lattice, program certification, protection, security, security classes

## 13 A type declaration and inference system for smalltalk

Alan H. Borning, Daniel H. H. Ingalls

January 1982 Proceedings of the 9th ACM SIGPLAN-SIGACT symposium on Principles

### of programming languages POPL '82

Publisher: ACM Press

Full text available: pdf(697.89 KB) Additional Information: full citation, abstract, references, citings

An experimental system for declaring and inferring type in Smalltalk is described. (In the current Smalltalk language, the programmer supplies no type declarations.) The system provides the benefits of type declaration in regard to compile-time checking and documentation, while still retaining Smalltalk's flexibility. A type hierarchy, which is integrated with the existing Smalltalk class hierarchy, allows one type to inherit the traits of another type. A type may also have parameters, which are ...

14 A framework for implementing pluggable type systems

Chris Andreae, James Noble, Shane Markstrum, Todd Millstein

October 2006 ACM SIGPLAN Notices, Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06, Volume 41 Issue 10

Publisher: ACM Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(294.23 KB)

Pluggable types have been proposed to support multiple type systems in the same programming language. We have designed and implemented JavaCOP, a program constraint system for implementing practical pluggable type systems for Java. JavaCOP enforces user-defined typing constraints written in a declarative and expressive rule language. We have validated our design by (re)implementing a range of type systems and program checkers. By using a program constraint system to implement p ...

**Keywords**: JavaCOP, pluggable type systems

15 Session II: Modular generic programming with extensible superclasses

Martin Sulzmann, Meng Wang

September 2006 Proceedings of the 2006 ACM SIGPLAN workshop on Generic programming WGP '06

**Publisher: ACM Press** 

Full text available: pdf(204.45 KB) Additional Information: full citation, abstract, references, index terms

"Generics for the Masses" (GM) and "Scrap your Boilerplate" (SYB) are generic programming approaches based on some inenious applications of Haskell type classes. To achieve modularity, the GM and SYB approach have been extended by using some experimental language extensions such as abstraction over type classes and recursive instances. Hence, the type class encodings behind the GM and SYB approach become less practical and harder to understand. We show that none of these type class features are n ...

Keywords: generic programming, type classes

16 Understanding class hierarchies using concept analysis

Gregor Snelting, Frank Tip

May 2000 ACM Transactions on Programming Languages and Systems (TOPLAS),

Volume 22 Issue 3

Publisher: ACM Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(433.91 KB)

A new method is presented for analyzing and reengineering class hierarchies. In our approach, a class hierarchy is processed along with a set of applications that use it, and a





fine-grained analysis of the access and subtype relationships between objects, variables, and class members is performed. The result of this analysis is again a class hierarchy, which is quaranteed to be behaviorally equivalent to the original hierarchy, but in which each object only contains the members that are req ...

**Keywords**: class hierarchy reengineering, concept analysis

17 Workshop papers: How secure is AOP and what can we do about it?

Bart De Win, Frank Piessens, Wouter Joosen

May 2006 Proceedings of the 2006 international workshop on Software engineering for secure systems SESS '06

Publisher: ACM Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(194.32 KB)

From a software engineering perspective, using Aspect-Oriented Programming (AOP) to build secure software has clear advantages. Until recently, the security perspective of this approach has been given less attention, however. This paper analyses the security risks in using AOP to develop secure software and discusses one particular solution to some of the identified risks, an aspect permission system. This permission system is one part of an overall AOP-based development platform for secure soft ...

**Keywords**: AOP, permission system, risks, security

18 Full papers: Runtime aspect weaving through metaprogramming

Jason Baker, Wilson Hsieh

April 2002 Proceedings of the 1st international conference on Aspect-oriented software development AOSD '02

Publisher: ACM Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(883.36 KB)

We describe an extension to the Java language, Handi-Wrap, that supports weaving aspects into code at runtime. Aspects in Handi-Wrap take the form of method wrappers, which allow aspect code to be inserted around method bodies like advice in AspectJ. Handi-Wrap offers several advantages over static aspect languages such as AspectJ. First, aspects can be woven into binary libraries. Second, a wrapper in Handi-Wrap is a firstclass Java value, which allows users to exploit Java mechanisms to defin ...

19 Jam---designing a Java extension with mixins

Davide Ancona, Giovanni Lagorio, Elena Zucca

September 2003 ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 25 Issue 5

**Publisher: ACM Press** 

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(1.33 MB) terms, review

In this paper we present Jam, an extension of the Java language supporting mixins, that is, parametric heir classes. A mixin declaration in Jam is similar to a Java heir class declaration, except that it does not extend a fixed parent class, but simply specifies the set of fields and methods a generic parent should provide. In this way, the same mixin can be instantiated on many parent classes, producing different heirs, thus avoiding code duplication and largely improving modularity and ...

Keywords: Java, language design

20 Integrating functional and imperative programming



David K. Gifford, John M. Lucassen

August 1986 Proceedings of the 1986 ACM conference on LISP and functional programming LFP '86

Publisher: ACM Press

Full text available: pdf(899.98 KB) Additional Information: full citation, references, citings

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 10

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc. Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player



Subscribe (Full Service) Register (Limited Service, Free) Login

Search: • The ACM Digital Library • O The Guide

target classes, enforcement point, insert bytecode

GEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: target classes enforcement point insert bytecode

Found 65,015 of 207,474

Sort results

by

Display results

relevance

expanded form

Save results to a Binder Search Tips

Open results in a new window

Try an Advanced Search Try this search in The ACM Guide

next

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 10

Best 200 shown

Relevance scale

Security and protection: Adaptiveness in well-typed Java bytecode verification

F. Y. Huang, C. B. Jay, D. B. Skillicorn

October 2006 Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research CASCON '06

Publisher: ACM Press

Full text available: pdf(301.36 KB) (1.80 KB)

Additional Information: full citation, abstract, references, index terms

Research on security techniques for Java bytecode has paid little attention to the security of the implementations of the techniques themselves, assuming that ordinary tools for programming, verification and testing are sufficient for security. However, different categories of security policies and mechanisms usually require different implementations. Each implementation requires extensive effort to test it and/or verify it. We show that programming with well-typed pattern structures in a statica ...

2 A static type system for JVM access control

Tomoyuki Higuchi, Atsushi Ohori

January 2007 ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 29 Issue 1

Publisher: ACM Press

Full text available: pdf(1.07 MB)

Full text available: pdf(317.30 KB) Additional Information: full citation, abstract, references, citings, index terms

A number of effective error detection tools have been built in recent years to check if a program conforms to certain design rules. An important class of design rules deals with sequences of events asso-ciated with a set of related objects. This paper presents a language called PQL (Program Query Language) that allows programmers to express such questions easily in an application-specific context. A query looks like a code excerpt corresponding to the shortest amount of code that would violate a ...

**Keywords**: SQL injection, bug finding, pattern matching, program traces, resource leaks, web applications

4 Programming languages: SCoPE: an AspectJ compiler for supporting user-defined



analysis-based pointcuts

Tomoyuki Aotani, Hidehiko Masuhara

March 2007 Proceedings of the 6th international conference on Aspect-oriented software development AOSD '07

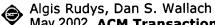
**Publisher: ACM Press** 

Full text available: pdf(195.33 KB) Additional Information: full citation, abstract, references, index terms

This paper proposes an approach called SCoPE, which supports user-defined analysisbased pointcuts in aspect-oriented programming (AOP) languages. The advantage of our approach is better integration with existing AOP languages than previous approaches. Instead of extending the language, SCoPE allows the programmer to write a pointcut that analyzes a program by using a conditional (if) pointcut with introspective reflection libraries. A compilation scheme automatically eliminates runtime tests fo ...

Keywords: AOP, analysis-based pointcuts, aspect-oriented programming languages, compiler design, point-cuts

Termination in language-based systems



May 2002 ACM Transactions on Information and System Security (TISSEC), Volume 5 Issue 2

Publisher: ACM Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(355.43 KB) terms

Language run-time systems are increasingly being embedded in systems to support runtime extensibility via mobile code. Such systems raise a number of concerns when the code running in such systems is potentially buggy or untrusted. Although sophisticated access controls have been designed for mobile code and are shipping as part of commercial systems such as Java, there is no support for terminating mobile code short of terminating the entire language run-time. This article presents a c ...

Keywords: Applets, Internet, Java, resource control, soft termination, termination

JRes: a resource accounting interface for Java

Grzegorz Czajkowski, Thorsten von Eicken October 1998 ACM SIGPLAN Notices, Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '98, Volume 33 Issue 10

Publisher: ACM Press

Additional Information:

Full text available: pdf(2.01 MB)

full citation, abstract, references, citings, index terms

With the spread of the Internet the computing model on server systems is undergoing several important changes. Recent research ideas concerning dynamic operating system extensibility are finding their way into the commercial domain, resulting in designs of extensible databases and Web servers. In addition, both ordinary users and service providers must deal with untrusted downloadable executable code of unknown origin and intentions. Across the board, Java has emerged as the language of choice fo ...

**Keywords**: Java, extensible systems, resource management

A specification of Java loading and bytecode verification

Allen Goldberg

November 1998 Proceedings of the 5th ACM conference on Computer and communications security CCS '98

**Publisher: ACM Press** 

Full text available: 🔁 pdf(1.15 MB) Additional Information: full citation, references, citings, index terms

Keywords: Java, bytecode verification, flow analysis, formal specification

Composing security policies with polymer

Lujo Bauer, Jay Ligatti, David Walker

June 2005 ACM SIGPLAN Notices, Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation PLDI '05, Volume 40 Issue 6

Publisher: ACM Press

Full text available: pdf(155.16 KB)

Additional Information: full citation, abstract, references, citings, index terms

We introduce a language and system that supports definition and composition of complex run-time security policies for Java applications. Our policies are comprised of two sorts of methods. The first is query methods that are called whenever an untrusted application tries to execute a security-sensitive action. A query method returns a suggestion indicating how the security-sensitive action should be handled. The second sort of methods are those that perform state updates as the pol ...

Keywords: composable security policies, edit automata, program monitors, run-time enforcement, security automata

Design and implementation of a distributed virtual machine for networked computers

Emin Gün Sirer, Robert Grimm, Arthur J. Gregory, Brian N. Bershad

December 1999 ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles SOSP

**'99**, Volume 33 Issue 5

Publisher: ACM Press

Full text available: pdf(1.62 MB)

Additional Information: full citation, abstract, references, citings, index

This paper describes the motivation, architecture and performance of a distributed virtual machine (DVM) for networked computers. DVMs rely on a distributed service architecture to meet the manageability, security and uniformity requirements of large, heterogeneous clusters of networked computers. In a DVM, system services, such as verification, security enforcement, compilation and optimization, are factored out of clients and located on powerful network servers. This partitioning of system fun ...

10 A practical comparison between Java and Ada in implementing a real-time

embedded system

Eric Potratz

December 2003 ACM SIGAda Ada Letters, Proceedings of the 2003 annual ACM SIGAda international conference on Ada: the engineering of correct and reliable software for real-time & distributed systems using ada and related technologies SigAda '03, Volume XXIV Issue 1

**Publisher: ACM Press** 

Full text available: pdf(259.58 KB)

Additional Information: full citation, abstract, references, citings, index terms, review

This paper presents a student's observations from an undergraduate research project that explored using Java to implement the software for a real-time embedded system that was originally implemented in a university-level real-time systems course using Ada 95. It briefly gives an overview of the project, the decision made concerning which Java virtual machine to use, and how that virtual machine performed in the real-time environment. It then goes into detail about the merits and drawbacks of usi ...

Keywords: Ada, Java, concurrency, conditional synchronization, drivers, embedded systems, memory management, object-oriented programming, package elaboration, performance, priority inversion, real-time systems, scheduling

11 A static type system for JVM access control

Tomoyuki Higuchi, Atsushi Ohori

August 2003 ACM SIGPLAN Notices, Proceedings of the eighth ACM SIGPLAN international conference on Functional programming ICFP '03, Volume 38 Issue 9

Publisher: ACM Press

Full text available: pdf(150.01 KB)

Additional Information: full citation, abstract, references, citings, index terms

This paper presents a static type system for JAVA Virtual Machine (JVM) code that enforces an access control mechanism similar to the one found, for example, in a JAVA implementation. In addition to verifying type consistency of a given JVM code, the type system statically verifies that the code accesses only those resources that are granted by the prescribed access policy. The type system is proved to be sound with respect to an operational semantics that enforces access control dynamically, si ...

Keywords: JVM, access control, stack inspection, type inference, type system

12 Kava: a Java dialect with a uniform object model for lightweight classes





David F. Bacon

June 2001 Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande JGI '01

**Publisher: ACM Press** 

Full text available: pdf(847.52 KB)

Additional Information: full citation, abstract, references, citings, index terms

Object-oriented programming languages have always distinguished between "primitive" and "user-defined" data types, and in the case of languages like C++ and Java, the primitives are not even treated as objects, further fragmenting the programming model. The distinction is especially problematic when a particular programming community requires primitive-level support for a new data type, as for complex, intervals, fixedpointed numbers, and so on.

We present Kav ...

13 Formalizing the safety of Java, the Java virtual machine, and Java card



Pieter H. Hartel, Luc Moreau

December 2001 ACM Computing Surveys (CSUR), Volume 33 Issue 4

**Publisher: ACM Press** 

Full text available: pdf(442.86 KB)

Additional Information: full citation, abstract, references, citings, index terms

We review the existing literature on Java safety, emphasizing formal approaches, and the impact of Java safety on small footprint devices such as smartcards. The conclusion is that although a lot of good work has been done, a more concerted effort is needed to build a coherent set of machine-readable formal models of the whole of Java and its implementation. This is a formidable task but we believe it is essential to build trust in Java safety, and thence to achieve ITSEC level 6 or Common Crite ...

Keywords: Common criteria, programming

14 Fine-grained interoperability through mirrors and contracts

Kathryn E. Gray, Robert Bruce Findler, Matthew Flatt

October 2005 ACM SIGPLAN Notices, Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications OOPSLA '05, Volume 40 Issue 10

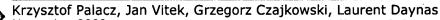
Publisher: ACM Press

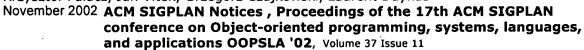
Full text available: pdf(391.61 KB) Additional Information: full citation, abstract, references, index terms

As a value flows across the boundary between interoperating languages, it must be checked and converted to fit the types and representations of the target language. For simple forms of data, the checks and coercions can be immediate; for higher order data, such as functions and objects, some must be delayed until the value is used in a particular way. Typically, these coercions and checks are implemented by an ad-hoc mixture of wrappers, reflection, and dynamic predicates. We observe that 1) the ...

Keywords: Java, contracts, interoperability, mirrors, scheme

15 Incommunicado: efficient communication for isolates





Publisher: ACM Press

Full text available: pdf(386.23 KB) Additional Information: full citation, abstract, references, citings

Executing computations in a single instance of safe language virtual machine can improve performance and overall platform scalability. It also poses various challenges. One of them is providing a fast inter-application communication mechanism. In addition to being efficient, such a mechanism should not violate any functional and non-functional properties of its environment, and should also support enforcement of application-specific security policies. This paper explores the design and implement ...

**Keywords**: application isolation, inter-application communication

16 Full papers: Runtime aspect weaving through metaprogramming

Jason Baker, Wilson Hsieh



**Publisher: ACM Press** 

Full text available: pdf(883.36 KB)

Additional Information: full citation, abstract, references, citings, index

We describe an extension to the Java language, Handi-Wrap, that supports weaving aspects into code at runtime. Aspects in Handi-Wrap take the form of method wrappers, which allow aspect code to be inserted around method bodies like advice in AspectJ. Handi-Wrap offers several advantages over static aspect languages such as AspectJ. First, aspects can be woven into binary libraries. Second, a wrapper in Handi-Wrap is a firstclass Java value, which allows users to exploit Java mechanisms to defin ...

17 Efficient control flow quantification

Christoph Bockisch, Sebastian Kanthak, Michael Haupt, Matthew Arnold, Mira Mezini October 2006 ACM SIGPLAN Notices, Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06, Volume 41 Issue 10

Publisher: ACM Press

Full text available: pdf(245.83 KB) Additional Information: full citation, abstract, references, index terms

Aspect-oriented programming (AOP) is increasingly gaining in popularity. However, the focus of aspect-oriented language research has been mostly on language design issues; efficient implementation techniques have been less popular. As a result, the performance of certain AOP constructs is still poor. This is in particular true for constructs that rely on dynamic properties of the execution (e.g., the cflow construct). In this paper, we present efficient implementation techniques for cfl ...

Keywords: aspect-oriented programming, control flow, virtual machine support

18 Secure program partitioning

Steve Zdancewic, Lantian Zheng, Nathaniel Nystrom, Andrew C. Myers August 2002 ACM Transactions on Computer Systems (TOCS), Volume 20 Issue 3

**Publisher: ACM Press** 

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(497.12 KB) terms

This paper presents secure program partitioning, a language-based technique for protecting confidential data during computation in distributed systems containing mutually untrusted hosts. Confidentiality and integrity policies can be expressed by annotating programs with security types that constrain information flow; these programs can then be partitioned automatically to run securely on heterogeneously trusted hosts. The resulting communicating subprograms collectively implement the original p ...

**Keywords**: Confidentiality, declassification, distributed systems, downgrading, integrity, mutual distrust, secrecy, security policies, type systems

19 Stack allocation and synchronization optimizations for Java using escape analysis

Jong-Deok Choi, Manish Gupta, Mauricio J. Serrano, Vugranam C. Sreedhar, Samuel P. Midkiff

November 2003 ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 25 Issue 6

Publisher: ACM Press



Full text available: pdf(632.85 KB) Additional Information: full citation, abstract, references, citings, index terms, review

This article presents an escape analysis framework for Java to determine (1) if an object is not reachable after its method of creation returns, allowing the object to be allocated on the stack, and (2) if an object is reachable only from a single thread during its lifetime, allowing unnecessary synchronization operations on that object to be removed. We introduce a new program abstraction for escape analysis, the connection graph, that is used to establish reachability relationshi ...

**Keywords**: Connection graphs, escape analysis, points-to graph

Mobile code: Empowering mobile code using expressive security policies



V. N. Venkatakrishnan, Ram Peri, R. Sekar

September 2002 Proceedings of the 2002 workshop on New security paradigms NSPW

Publisher: ACM Press

Full text available: pdf(853.33 KB)

Additional Information: full citation, abstract, references, citings, index

Existing approaches for mobile code security tend to take a conservative view that mobile code is inherently risky, and hence focus on confining it. Such confinement is usually achieved using access control policies that restrict mobile code from taking any action that can potentially be used to harm the host system. While such policies can be helpful in keeping "bad applets" in check, they preclude a large number of useful applets. We therefore take an alternative view of mobile code security, ...

Keywords: code transformation, mobile code security, security policies

Results 1 - 20 of 200 Result page: 1 2 3 4 5 6 7 8 9 10

The ACM Portal is published by the Association for Computing Machinery. Copyright @ 2007 ACM, Inc. Terms of Usage Privacy Policy Code of Ethics Contact\_Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player



Subscribe (Full Service) Register (Limited Service, Free) Login

Search: © The ACM Digital Library

C The Guide

java, enforcement code

SEARCH

THE ACM DIGHTAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: java enforcement code

Found 15,050 of 207,474

Sort results

by Display

results

relevance

expanded form

Save results to a Binder Search Tips

Open results in a new

Try an Advanced Search Try this search in The ACM Guide

Results 1 - 20 of 200

window

Result page: 1 2 3 4 5 6 7 8 9 10

next Relevance scale

Security issues surrounding programming languages for mobile code: JAVA vs. Safe-



<u>Tcl</u>

Best 200 shown

Stefanos Gritzalis, George Aggelis

April 1998 ACM SIGOPS Operating Systems Review, Volume 32 Issue 2

Publisher: ACM Press

Full text available: pdf(1.42 MB)

Additional Information: full citation, abstract, references

JAVA is claimed to be a system programming language having a number of advantages over traditional programming languages. These advantages stem from the fact that it is a platform - independent language, thus promising truly network oriented computing as long as a nearly universal system for distributing applications. On the other hand, although being an interpreted, much simpler, scripting language, Safe-Tcl was proposed as an executable contents type of MIME and thus as the standard language f ...

2 Programming for separation of concerns (PSC): Policy-driven reflective enforcement





of security policies

Ian Welch, Fan Lu

April 2006 Proceedings of the 2006 ACM symposium on Applied computing SAC '06

**Publisher: ACM Press** 

Full text available: pdf(105.96 KB) Additional Information: full citation, abstract, references, index terms

Practical experience has shown that separating security enforcement code from functional code using separation of concerns techniques such as behavioural reflection leads to improvements in code undestandability and maintainability. However, using these techniques at requires providing a consistent and declarative way to specify policies. We have developed a prototype tool that allows the use of Ponder policies that are enforced by the Kava metaobject protocol. This prototype translates high-lev ...

Keywords: reflection, security policies

Security and protection: Adaptiveness in well-typed Java bytecode verification



F. Y. Huang, C. B. Jay, D. B. Skillicorn October 2006 Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research CASCON '06

Publisher: ACM Press

Full text available: pdf(301.36 KB) Additional Information: full citation, abstract, references, index terms

### htm(1.80 KB)

Research on security techniques for Java bytecode has paid little attention to the security of the implementations of the techniques themselves, assuming that ordinary tools for programming, verification and testing are sufficient for security. However, different categories of security policies and mechanisms usually require different implementations. Each implementation requires extensive effort to test it and/or verify it. We show that programming with well-typed pattern structures in a statica ...

Mobile code: Empowering mobile code using expressive security policies



V. N. Venkatakrishnan, Ram Peri, R. Sekar

September 2002 Proceedings of the 2002 workshop on New security paradigms NSPW

**Publisher: ACM Press** 

Full text available: pdf(853.33 KB)

Additional Information: full citation, abstract, references, citings, index

Existing approaches for mobile code security tend to take a conservative view that mobile code is inherently risky, and hence focus on confining it. Such confinement is usually achieved using access control policies that restrict mobile code from taking any action that can potentially be used to harm the host system. While such policies can be helpful in keeping "bad applets" in check, they preclude a large number of useful applets. We therefore take an alternative view of mobile code security, ...

Keywords: code transformation, mobile code security, security policies

Separating access control policy, enforcement, and functionality in extensible





systems

Robert Grimm, Brian N. Bershad

February 2001 ACM Transactions on Computer Systems (TOCS), Volume 19 Issue 1

Publisher: ACM Press

Full text available: pdf(164.03 KB)

Additional Information: full citation, abstract, references, citings, index terms, review

Extensible systems, such as Java or the SPIN extensible operating system, allow for units of code, or extensions, to be added to a running system in almost arbitrary fashion. Extensions closely interact through low-latency but type-safe interfaces to form a tightly integrated system. As extensions can come from arbitrary sources, not all of whom can be trusted to conform to an organization's security policy, such structuring raises the question of how security constraints are enforced in an ...

Keywords: Java, SPIN, access check, auditing, extensible systems, policy-neutral enforcement, protection domain, protection domain transfer, security policy

Safety critical systems: A type system to assure scope safety within safety-critical





Java modules

Kelvin Nilsen

October 2006 Proceedings of the 4th international workshop on Java technologies for real-time and embedded systems JTRES '06

Publisher: ACM Press

Full text available: pdf(409.03 KB) Additional Information: full citation, abstract, references, index terms

To address the needs of safety-critical system developers, a type system based on Java 5.0 meta-data annotations and special bytecode verification techniques is described. This type system enables programmers to develop code for which the byte code verifier is able to prove the absence of scoped memory protocol errors, thereby eliminating the need for run-time assignment checks. Benefits of the type system include improved software reliability, easier maintenance and integration of independently ...

Keywords: DO-178B, RTSJ, java, safety-critical certification, scoped memory

Model-carrying code: a practical approach for safe execution of untrusted



applications

R. Sekar, V.N. Venkatakrishnan, Samik Basu, Sandeep Bhatkar, Daniel C. DuVarney October 2003 ACM SIGOPS Operating Systems Review, Proceedings of the nineteenth ACM symposium on Operating systems principles SOSP '03, Volume 37 Issue

**Publisher: ACM Press** 

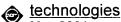
Additional Information: full citation, abstract, references, citings, index Full text available: pdf(301.30 KB)

This paper presents a new approach called model-carrying code (MCC) for safe execution of untrusted code. At the heart of MCC is the idea that untrusted code comes equipped with a concise high-level model of its security-relevant behavior. This model helps bridge the gap between high-level security policies and low-level binary code, thereby enabling analyses which would otherwise be impractical. For instance, users can use a fully automated verification procedure to determine if the code ...

**Keywords**: mobile code security, policy enforcement, sand-boxing, security policies

Secure virtual enclaves: Supporting coalition use of distributed application





May 2001 ACM Transactions on Information and System Security (TISSEC), Volume 4 Issue 2

Publisher: ACM Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(462.10 KB)

The Secure Virtual Enclaves (SVE) collaboration infrastructure allows multiple organizations to share their distributed application objects, while respecting organizational autonomy over local resources. The infrastructure is transparent to applications, which may be accessed via a web server, or may be based on Java or Microsoft's DCOM. The SVE infrastructure is implemented in middleware, with no modifications to COTS operating systems or network protocols. The system enables dynamic updates to ...

Keywords: Access control, coalition, collaborative system, group communication, middleware, security policy

SASI enforcement of security policies: a retrospective



Ulfar Erlingsson, Fred B. Schneider

September 1999 Proceedings of the 1999 workshop on New security paradigms NSPW

Publisher: ACM Press

Full text available: Top pdf(862.14 KB) Additional Information: full citation, references, citings, index terms

10 Termination in language-based systems Algis Rudys, Dan S. Wallach





May 2002 ACM Transactions on Information and System Security (TISSEC), Volume 5 Issue 2

Publisher: ACM Press

Full text available: pdf(355.43 KB)

Additional Information: full citation, abstract, references, citings, index terms

Language run-time systems are increasingly being embedded in systems to support runtime extensibility via mobile code. Such systems raise a number of concerns when the code running in such systems is potentially buggy or untrusted. Although sophisticated access controls have been designed for mobile code and are shipping as part of commercial systems such as Java, there is no support for terminating mobile code short of terminating the entire language run-time. This article presents a c ...

Keywords: Applets, Internet, Java, resource control, soft termination, termination

11 A protection scheme for mobile agents on Java

D. Hagimont, L. Ismail

September 1997 Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking MobiCom '97

Publisher: ACM Press

Full text available: pdf(1.10 MB)

Additional Information: full citation, references, citings, index terms

12 Access rights analysis for Java

Larry Koved, Marco Pistoia, Aaron Kershenbaum

November 2002 ACM SIGPLAN Notices, Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '02, Volume 37 Issue 11

Publisher: ACM Press

Full text available: pdf(360.93 KB)

Additional Information: full citation, abstract, references, citings, index terms

Java 2 has a security architecture that protects systems from unauthorized access by mobile or statically configured code. The problem is in manually determining the set of security access rights required to execute a library or application. The commonly used strategy is to execute the code, note authorization failures, allocate additional access rights, and test again. This process iterates until the code successfully runs for the test cases in hand. Test cases usually do not cover all paths th ...

**Keywords**: Java security, access rights, call graph, data flow analysis, invocation graph, security

13 Systems and prototypes: Java support for data-intensive systems: experiences



building the telegraph dataflow system

Mehul A. Shah, Michael J. Franklin, Samuel Madden, Joseph M. Hellerstein December 2001 ACM SIGMOD Record, Volume 30 Issue 4

Publisher: ACM Press

Full text available: pdf(1.38 MB)

Additional Information: full citation, abstract, references, citings

Database system designers have traditionally had trouble with the default services and interfaces provided by operating systems. In recent years, developers and enthusiasts have increasingly promoted Java as a serious platform for building data-intensive servers. Java provides a number of very helpful language features, as well as a full run-time environment reminiscent of a traditional operating system. This combination of features and community support raises the question of whether Java is be ...

14 Migration: Luna: a flexible Java protection system

Chris Hawblitzel, Thorsten von Eicken

December 2002 ACM SIGOPS Operating Systems Review, Volume 36 Issue SI

Publisher: ACM Press

Full text available: pdf(1.39 MB)

Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>cited by</u>, <u>index</u> <u>terms</u>

Extensible Java systems face a difficult trade-off between sharing and protection. On one hand, Java's ability to run different protection domains in a single virtual machine enables domains to share data easily and communicate without address space switches. On the other hand, unrestricted sharing blurs the boundaries between protection domains, making it difficult to terminate domains and enforce restrictions on resource usage. Existing solutions to these problems restrict sharing in an ad-hoc ...

15 <u>Software engineering: Propagation of JML non-null annotations in Java programs</u>

Maciej Cielecki, J□drzej Fulara, Krzysztof Jakubczyk, Łukasz Jancewicz

August 2006 Proceedings of the 4th international symposium on Principles and practice of programming in Java PPPJ '06

**Publisher: ACM Press** 

Full text available: pdf(376.78 KB) Additional Information: full citation, abstract, references, index terms

Development of high quality code is notably difficult. Tools that help maintaining the proper quality of code produced by programmers can be very useful: they may increase the quality of produced software and help managers to ensure that the product is ready for the market. One of such tools is ESC/Java2, a static checker of Java Modeling Language annotations. These annotations can be used to ensure that a certain assertion is satisfied during the execution of the program, among the others - to ...

16 Computability classes for enforcement mechanisms

Kevin W. Hamlen, Greg Morrisett, Fred B. Schneider

January 2006 ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 28 Issue 1

Publisher: ACM Press

Full text available: 1 pdf(337.62 KB)

Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index</u> terms

A precise characterization of those security policies enforceable by program rewriting is given. This also exposes and rectifies problems in prior work, yielding a better characterization of those security policies enforceable by execution monitors as well as a taxonomy of enforceable security policies. Some but not all classes can be identified with known classes from computational complexity theory.

**Keywords**: Program rewriting, edit automata, execution monitoring, inlined reference monitoring, reference monitors, security automata

17 Analysis against attacks: Using web application construction frameworks to protect

against code injection attacks
Benjamin Livshits, Úlfar Erlingsson

June 2007 Proceedings of the 2007 workshop on Programming languages and analysis for security PLAS '07

Publisher: ACM Press

Full text available: pdf(566.79 KB) Additional Information: full citation, abstract, references, index terms

In recent years, the security landscape has changed, with Web applications vulnerabilities becoming more prominent that vulnerabilities stemming from the lack of type safety, such

as buffer overruns. Many reports point to code injection attacks such as cross-site scripting and RSS injection as being the most common attacks against Web applications to date. With Web 2.0 existing security problems are further exacerbated by the advent of Ajax technology that allows one to create and compose HTM ...

**Keywords**: code injection attacks, same-origin policy, software construction frameworks, software security

18 The KaffeOS Java runtime system

Godmar Back, Wilson C. Hsieh

July 2005 ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 27 Issue 4

Publisher: ACM Press

Full text available: pdf(704.30 KB)

Additional Information: full citation, abstract, references, citings, index terms, review

Single-language runtime systems, in the form of Java virtual machines, are widely deployed platforms for executing untrusted mobile code. These runtimes provide some of the features that operating systems provide: interapplication memory protection and basic system services. They do not, however, provide the ability to isolate applications from each other. Neither do they provide the ability to limit the resource consumption of applications. Consequently, the performance of current systems degra ...

**Keywords**: Robustness, garbage collection, isolation, language runtimes, resource management, termination, virtual machines

19 A practical type system and language for reference immutability

Adrian Birka, Michael D. Ernst

October 2004 ACM SIGPLAN Notices, Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04, Volume 39 Issue 10

**Publisher: ACM Press** 

Full text available: pdf(171.73 KB)

Additional Information: full citation, abstract, references, citings, index terms

This paper describes a type system that is capable of expressing and enforcing immutability constraints. The specific constraint expressed is that the abstract state of the object to which an immutable reference refers cannot be modified using that reference. The abstract state is (part of) the transitively reachable state: that is, the state of the object and all state reachable from it by following references. The type system permits explicitly excluding fields or objects from the abstract ...

Keywords: Javar, Javari, const, immutability, mutable, readonly, type system, verification

20 MOCA: a service framework for mobile computing devices

James Beck, Alain Gefflaut, Nayeem Islam

August 1999 Proceedings of the 1st ACM international workshop on Data engineering for wireless and mobile access MobiDe '99

**Publisher: ACM Press** 

Full text available: pdf(911.37 KB) Additional Information: full citation, references, citings, index terms

**Keywords**: Java, component software, mobile device, service discovery, service

framework

Results 1 - 20 of 200

Result page:  $1 \quad \underline{2} \quad \underline{3} \quad \underline{4} \quad \underline{5} \quad \underline{6} \quad \underline{7} \quad \underline{8} \quad \underline{9} \quad \underline{10}$ 

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc. Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	. 301	726/21.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:36
S2	234	719/320.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:36
S3	534	S1 S2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:41
S4	0	S3 and (authoriz\$3 same enforce\$4 same point\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR .	ON	2007/08/08 18:41
S5	128	S3 and java	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:41
S6	265	(authoriz\$4 same enforce\$4 same point\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:42
S7	62	(authoriz\$4 same enforce\$4 same point\$2) and java	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:42

S8	160913	(authoriz\$7 adj enforc\$4) samwe framework and application\$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:42
`S9		(authoriz\$7 adj enforc\$4) same framework and application\$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:43
S10	2	((authorization adj enforcement) same framework) and application\$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:44
S11	202	713/152.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:44
S12	733	713/152.ccls. S3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:44
S13	2	S12 and(authoriz\$3 same enforce\$4 same point\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:45
S14	13	(authorization\$2 adj enforcement\$2) and framework\$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:45

	_					
S15	2	((authorization same enforcement\$2) same ((pluggable or insert or inject) near10 code)) and framework\$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:46
S16	5	((authorization same enforcement same ((pluggable or insert or inject) near10 code)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:47
S17	0	(authorization\$2 same enforcment same ((pluggable or insert or inject) near4 code) same runtime)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:47
S18	1	(authorization adj enforcement and ((pluggable or insert or inject) adj4 code) and runtime)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:48
S19	4	(java near4 (bytecode or scriptcode or (byte adj code) or (script adj code)) near4 (insert\$4 or inject\$4)) same runtime	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:50
S20	9	(java same (bytecode or scriptcode or (byte adj code) or (script adj code)) near4 (insert\$4 or inject\$4)) same runtime	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 18:51
S21	2	(java same (bytecode or scriptcode or (byte adj code) or (script adj code)) same (insert\$4 or inject\$4) same (runtime or (class adj loader)) same security)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:03

<del></del>			<u> </u>	Γ		1
S22	4523	(local adj variable)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR .	ON	2007/08/08 19:04
S23	197	(local adj variable) and (instance adj variable)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JOO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:04
S24		(local adj variable) and (instance adj variable) and (permission adj class\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:07
S25	2	(authorization adj enforcement adj point)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:08
S26	<b>72</b>	permission\$4 adj class and (java)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:21
S27	1806	(target adj class\$3)	US-PGUB; USPA USOCK; FPRS; EPO; JPO; DERWENT; IBM_TDB		ON	2007/08/08 19:21
S28	. 227	(target adj class\$3) and java	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:22

S29 .	6844	application\$2 with authorizat\$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:23
S30	13978	application\$2 with authoriz\$8	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:23
S31	3262	application\$2 with authoriz\$8 and java	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:23
S32	0	application\$2 with authoriz\$8 and java and (class44)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	Ο̈́Ν	2007/08/09 11:39
S33	1955	application\$2 with authoriz\$8 and java and (class\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:24
S34	78	application\$2 with authoriz\$8 and java and (class\$2) and (insert\$4 inject\$4) with (class\$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/08 19:24
S35	452	(bytecode (byte adj code)) same security	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 11:39

Cac	35	C2F come notice:	LIC DCDLID	OB	001	2007/09/00 11:20
S36	35	S35 same policy	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 11:39
S37	14	(bytecode (byte adj code)) with (insert\$4 inject\$4) same security	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 11:40
S38	52	java same security same permission with class	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 14:11
S39	1326	713/189.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR .	ON	2007/08/09 14:33
S40	2	713/189.ccls. and (authoriz\$3 same enforce\$4 same point\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 14:33
S41	0	713/189.ccls. and (authoriz\$4 same enforce\$4 same point\$2) and java	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR .	ON	2007/08/09 14:34
S42	10	713/189.ccls. and application\$2 with authoriz\$8 and java and (class\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 14:57

S43	2040	713/189.ccls. 726/21.ccls. 719/320. ccls. 713/152.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 14:58
S44	1	S43 and (authorization adj enforcement adj point\$2).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 14:59
S45	. 5	S43 and (permission\$2 near2 class\$2).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:00
S46	1	S43 and (permission\$2 near2 class\$2).clm. and (insert\$4 inject\$4).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:01
S47	8	KILROY near3 JOHN.in.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:02
S48	67409	INTERNATIONAL adj BUSINESS adj MACHINES adj CORPORATION.as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:03
S49	6	INTERNATIONAL adj BUSINESS adj MACHINES adj CORPORATION.as. and (permission\$2 near2 class\$2). clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:14

S50	5012	java.clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:14
S51	1194	java.clm. and (class).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:14
S52	513	java.clm. and (class).clm. and ((byte adj code) bytecode)"clm."	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:15
S53	513	java.clm. and (class).clm. and ((byte adj code) bytecode)"clm."	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:15
S54	159	java.clm. and (class).clm. and ((byte adj code) bytecode).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:15
S55	7	java.clm. and (class).clm. and ((byte adj code) bytecode).clm. and (declaration).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/09 15:15